World Wide Web
DOI 10.1007/s11280-012-0179-z

Riding the tide of sentiment change: sentiment analysis with evolving online reviews

Yang Liu · Xiaohui Yu · Aijun An · Xiangji Huang

Received: 21 December 2011 / Revised: 1 May 2012 /
Accepted: 25 June 2012

Abstract The last decade has seen a rapid growth in the volume of online reviews. A great deal of research has been done in the area of opinion mining, aiming at analyzing the sentiments expressed in those reviews towards products and services. Most of the such work focuses on mining opinions from a collection of reviews posted during a particular period, and does not consider the change in sentiments when the collection of reviews evolve over time. In this paper, we fill in this gap, and study the problem of developing adaptive sentiment analysis models for online reviews. Given the success of latent semantic modeling techniques, we propose two adaptive methods to capture the evolving sentiments. As a case study, we also investigate the possibility of using the extracted adaptive patterns for sales prediction. Our proposal is evaluated on an IMDB dataset consisting of reviews of selected movies and their box office revenues. Experimental results show that the adaptive methods can capture sentiment changes arising from newly available reviews, which helps greatly improve the prediction accuracy.

This paper is a substantial extension of the conference version: Xiaohui Yu, Yang Liu, Aijun An: An Adaptive Model for Probabilistic Sentiment Analysis. Web Intelligence 2010: 661–667.

Y. Liu · X. Yu (✉)
School of Computer Science and Technology, Shandong University, Jinan 250101, China
e-mail: xyu@sdu.edu.cn

Y. Liu
e-mail: yliu@sdu.edu.cn

X. Yu · X. Huang
School of Information Technology, York University, Toronto, ON, Canada, M3J 1P3

X. Huang
e-mail: jhuang@yorku.ca

A. An
Department of Computer Science and Engineering, York University,
Toronto, ON, Canada, M3J 1P3
e-mail: aan@cse.yorku.ca

Published online: 18 July 2012

Keywords sentiment analysis · adaptive algorithm · opinion mining

1 Introduction

With the rapid advancement of Web 2.0 technologies, which facilitate people to contribute rather than simply receive information, a large amount of review texts are generated and become available online. These user-generated opinion-rich contents are credible sources of knowledge that can not only help users make better judgements but assist manufacturers of products in keeping track of customer sentiments. In fact, online reviews have been shown to be second only to word-out-mouth in a study that compares the factors influencing purchase decisions [31]. Therefore, online reviews can be very valuable, as collectively such reviews reflect the "wisdom of crowds" and can be a good indicator of a product's future sales performance. However, with tens and thousands of reviews being generated everyday on almost everything, e.g., sellers, products, and services, at various websites such as CNet and Epinions, it has become increasingly difficult for an individual to manually collect and digest the reviews of his/her interest. As such, opinion mining has become an active area of research in the past few years, and has produced some important results.

The work presented in this paper falls into the domain of opinion mining, and can be considered as an effort along the general direction of Wisdom Web of Things (W2T). W2T has recently emerged as an important new data cycle to integrate social, physical, and cyber worlds [39–41]. In contrast to most existing Web and ubiquitous computing technologies, which are only limited to specific data or applications, A W2T data cycle system is designed to perform a complete study "from things to data, information, knowledge, wisdom, services, humans, and then back to things". Our work is closely related to W2T in that (1) opinion mining and online review analysis are important applications of Web Intelligence, which is a major component of intelligence in the hyper world [41]; and (2) we attempt to integrate people's opinions (in the social world), product sales (in the physical world), and computer systems (in the cyber world) into an entity, which represents a direct application of the W2T data cycle in the hyper world.

Some studies in opinion mining attempt to answer the question of whether the polarity and the volume of reviews that are available online have a significant effect on actual customer purchasing [1, 8, 16, 17]. Various economic functions have been utilized to examine the relationship between opinions discovered from product reviews and revenue growth, stock trading volume change, as well as the bidding price variation in commercial Websites, such as eBay [4, 11, 29]. In particular, Gruhl et al. [16] show that the volume of relevant postings can help predict the sales ranking of books on Amazon, especially the spikes in ranking. In contrast to the above work which captures sentiments with explicit rating indication such as the number of stars, there are also a few studies that attempt to exploit text mining strategies for sentiment understanding. For example, Ghose and Ipeirotis [12, 13] demonstrate that the reviews can have an impact on sales performance, and review texts contain rich information that cannot be easily captured using numerical ratings. Liu et al. [23] study the important problem of utilizing the extracted sentiment patterns for predicting future product sales. They observe that simply classifying reviews as

positive or negative, as most current sentiment-mining approaches are designed for, does not provide a comprehensive explanation of the sentiments reflected in reviews. Therefore, they propose a probabilistic model called Sentiment PLSA (S-PLSA for short) based on the assumption that sentiment consists of multiple hidden aspects.

It is worth noting that the sentiments toward a product or service may evolve over time. For example, the service of a hotel can improve or degrade over time, and a product may improve its features and functionalities with upgrades. Such changes may be reflected in newly available reviews. Accordingly, a sentiment analysis system must adjust itself and capture the changes effectively. Unfortunately, most of the previous work fails to address this issue, focusing only on analyzing the sentiments during a particular period in a batch fashion. In this paper, we fill in this gap, and investigate the important problem of constructing adaptive sentiment analysis models.

Given the success of latent semantic modeling in opinion mining, we propose two adaptive methods, both of which extend the S-PLSA model by automatically adjusting the sentiment system when new reviews become available. To improve the efficiency of building/maintaining the S-PLSA models, we develop two algorithms to update the model parameters without invoking overhaul re-construction. First, we notice that if the number of reviews that the model is built on is sufficiently large, the addition of a new review is unlikely to have significant influence on parameter estimation. Therefore, when a new review becomes available, it is possible to update the parameters incrementally (albeit approximately) without invoking reconstruction of the model. Based on this observation, we develop our first method, which can be considered as a simplified method for parameter estimation in the EM algorithm. This method provides light-weight parameter estimation, and can be applied when efficiency is of major concern. In contrast, the second method we propose takes a Bayesian approach. It is motivated by the principle of quasi-Bayesian (QB) estimation, which has found successful applications in various domains such as adaptive speech recognition and text retrieval [6]. One salient feature of this modeling is the judicious use of hyperparameters, which can be recursively updated in order to obtain up-to-date posterior distributions and to estimate new model parameters. With its solid statistical foundation, we expect the second method to provide more accurate estimation of the parameters than the first one, making it a better choice when accuracy is more important.

Since our objective of conducting sentiment analysis is to analyze reviews and distill useful knowledge that could be of economic value, we also investigate the relationship between sentiments and sales performance. To this end, we use a model called ARSA (which stands for Auto-Regressive Sentiment-Aware) to quantitatively measure this connection [23], and opt to implement our methods to predict future box office revenues in the movie domain. We design measures to preprocess the time series of box office revenues, and show that the methods proposed can be extended to many other domains. We also experimentally evaluate each of the S-PLSA based adaptive learning methods, and study how they fare against other possible alternatives.

The rest of the paper is organized as follows. In Section 2, we provide a review of related work. In Section 3, we start with presenting the algorithm of S-PLSA for sentiment analysis. In Section 4, we describe the general framework for adaptive sentiment analysis of reviews, and elaborate the two proposed methods. We introduce

our method of using the proposed models for product sales prediction in the movie domain in Section 5, and report the experimental results on a movie review dataset in Section 6. Section 7 concludes this paper.

2 Related work

To the best of our knowledge, few study has addressed the problem of developing adaptive sentiment analysis models. But there are several lines of related work which we will review in this section.

2.1 Sentiment mining

With more and more users becoming comfortable with the Web, a large number of people are writing reviews online. Consequently, the number of reviews grows rapidly. When trying to locate information on a product, a general Web search would retrieve a large collection of documents; however getting an overall sense of the reviews can be daunting and time-consuming. To solve these problems, recent years have seen a growing interest in sentiment mining, whose objective is to find opinions, feelings, and attitude expressed in text, rather than facts. In the literature, sentiment mining also goes under various names, such as opinion mining [7, 15, 24], sentiment analysis [25, 26, 35], etc. Its related work may come from both computer science and linguistics, and its immediate applications may involve data mining, market intelligence, and customer relationship management.

The task of sentiment analysis can be roughly divided into three sub-categories: determining subjectivity [34, 36, 37], determining orientation, and determining the strength of orientation [33, 35], and most of the studies focus on investigating the sentiment orientation of words, phrases, and documents. Sentiment classification is usually defined as the problem of binary classification of a document or a sentence. In some recent work, Kamps and Marx [20] tried to evaluate the semantic distance from a word to good/bad with WordNet. They first defined a graph on the adjectives appeared in both the WordNet and the target term list. If two adjectives in WordNet display a synonymy relation, a link will be added between them. In turn, the semantic orientation of a word w is decided by its relative distance to *good* and *bad*. Pang et al. [28] employed three machine learning approaches (Naive Bayes, Maximum Entropy, and Support Vector Machine) to label the polarity of IMDB movie reviews. They represented reviews in several formats, where the unigram representation was the simplest but the most successful one. In addition, the Support Vector Machines yielded the best result among three classifiers in their experiments. Aside from the explicit two-class classification problem, Pang and Lee [27] and Zhang and Varadarajan [38] tended to determine the author's opinion with different rating scales (i.e., the number of stars). Further, a metric labeling approach was designed to compare with both multi-class and regression versions of Support Vector Machines. Liu et al. [21] built a framework to compare consumer opinions of competing products using multiple feature dimensions. After deducting supervised rules from product reviews, the strength and weakness of the product were visualized with an *Opinion Observer*. Snyder and Barzilay [32] improved aspect level rating prediction by modeling the dependent relation between various aspects. Observing that simply

classifying reviews as being positive or negative, as most of the previous work is designed for, does not provide a comprehensive understanding of sentiments reflected, Liu et al. [23] assumed that sentiment consists of multiple hidden aspects, and used a probability model to quantitatively measure the relationship between sentiment aspects and reviews.

2.2 Applications in business intelligence

Academics have recognized the impact of online reviews on business intelligence, and have produced some important results in this area. Among them, some studies attempt to answer the question of whether the polarity and the volume of reviews available online have a measurable and significant effect on actual customer purchasing [1, 5, 9, 11, 16, 17, 29, 42]. To this end, most studies use some form of hedonic regression [30] to analyze the significance of different features to certain function (e.g., measuring the utility to the the consumer). Various economic functions have been utilized in the field of examining revenue growth, stock trading volume change, as well as the bidding price variation in commercial Websites, such as Amazon and eBay. In most of the studies cited above, the review sentiment was captured by explicit rating indication such as the number of stars, but only a few studies attempted to exploit text mining strategies for sentiment classification. To fill in this gap, Ghose and Ipeirotis [12] claimed that review texts contain richer information that cannot be easily captured using simple numerical ratings. In their study, they assigned a "dollar value" to a collection of adjective-noun pairs, as well as adverb-verb pairs, and investigated how they affect the bidding prices of various products at Amazon.

2.3 Adaptive probabilistic latent semantic modeling

Latent semantic modeling has become very popular as a completely unsupervised technique for topic discovery in large documents. These models, such as PLSA [18] and LDA [3], exploit co-occurrence patterns of words in documents to understand semantically meaningful probabilistic clusters of words. These models assign a probabilistic membership to documents in the latent topic space, assisting us for viewing and processing the data in a lower-dimensional space. PLSA was shown to be a special variant of LDA with a uniform Dirichlet prior in a maximum a posterior model [14], and has been successfully applied to content-based recommendation and collaborative filtering [2, 19, 22]. However, one limitation of the model is its incapacity of adapting itself as new data become available, and the problem will get worse when the data arrive in a stream. This is due to the fact that the PLSA model is estimated only for documents that appear in the training set, and re-training model using both existing training data and new data from scratch is highly inefficient. Motivated by the idea of quasi-Bayes estimate, Chien and Wu [6] propose an incremental learning method to estimate the model parameters by maximizing an approximate posterior distribution, and expect that such an approach can effectively absorb the domain knowledge from the newly arrived data. Here, we adapt their methodology in one of our two adaptive models, and explore the possibility of developing adaptive models for predicting product sales using sentiments that dynamically change as new online reviews come in.

3 S-PLSA

Many existing models and algorithms for sentiment mining are developed for the binary classification problem, i.e., to classify the sentiment of a review as positive or negative. However, a common deficiency of the work is that the proposed approaches usually attempt to extract only the overall sentiments of a review, but can not distinguish different aspects within a sentiment, such as polarity, orientation, graduation, etc. Meanwhile, a general classification of good or bad is not very informative to the reader, who always seeks to dig into different facets and explore more detailed opinions. All these concerns call for a model that can have a more in-depth analysis of the multi-facets nature of review sentiments.

To this end, Liu et al. [23] propose the S-PLSA model, in which a review can be considered as being generated under the influence of a number of hidden sentiment factors. Inspired by the PLSA model [18, 19], the use of hidden factors in S-PLSA provides the ability to accommodate the intricate nature of sentiments, with each hidden factor focusing on one specific aspect. What differentiates S-PLSA from conventional PLSA is its use of a set of appraisal words [35] as the basis for feature representation. In order to represent a given review as an input, S-PLSA computes the (relative) frequencies of various words in a review and use the resulting multidimensional feature vector as the representation of the document. In particular, instead of adopting the frequencies of all words appearing in reviews, the model focuses on the set of *appraisal words* extracted from an appraisal lexicon [35]. The rationale is that those appraisal words, such as "*good*" or "*terrible*", are more indicative of the review's sentiments than other words. As a concrete example of appraisal words, the lexical entry for the appraisal word *beautiful* can be described as follows:

```
beautiful
Attitude:     appreciation/reaction-quality
Orientation:  positive
Force:        neutral
Focus:        neutral
Polarity:     unmarked
```

where the adjective is fully described with four types of attributes. In this context, the attitude of an appraisal word provides the appraisal expressed as either *affect*, *appreciation*, or *judgment*. Orientation describes whether the appraisal is *positive* or *negative*. Graduation presents the intensity of appraisal wrt. two independent folds of *force* and *focus*. Finally, polarity is *marked* if it is confined with a polarity marker (such as 'not'), or *unmarked* otherwise [35].

Now we formally describe the use of S-PLSA to extract the hidden sentiments of reivews. For a given set of N reviews $\mathcal{D} = \{d_1, \ldots, d_N\}$, and the set of M appraisal words $\mathcal{W} = \{w_1, \ldots, w_M\}$, the S-PLSA model dictates that the joint probability of observed pair (d_i, w_j) is generated by

$$P(d_i, w_j) = P(d_i) \sum_{k=1}^{K} P(w_j|z_k) P(z_k|d_i), \tag{1}$$

where $z_k \in \mathcal{Z} = \{z_1, \ldots, z_K\}$ corresponds to the latent sentiment factor, and where we assume that d_i and w_j are independent conditioned on the mixture of associated sentiment factor z_k. The set of parameters θ of this model consist of

$\{P(w_j|z_k), P(z_k|d_i)\}$, where $\sum_{j=1}^{M} P(w_j|z_k) = 1$ and $\sum_{k=1}^{K} P(z_k|d_i) = 1$, and there totally exist $KM + KN$ probabilities in θ. If we consider the number $c(d_i, w_j)$ of word w_j occurring in document d_j and accumulate the log likelihood of training data $X = \{d_i, w_j\}$ using θ, then

$$\log P(X|\theta) = \sum_{i=1}^{n} \sum_{j=1}^{M} c(d_i, w_j) \log P(d_i, w_j). \tag{2}$$

S-PLSA parameter set θ thus can be found by maximizing the accumulated log likelihood

$$\theta_{ML} = \arg\max_{\theta} \log P(X|\theta). \tag{3}$$

As the hidden parameter z_k is embedded in the above function, the expectation-maximization (EM) algorithm [18, 19] can be adopted to estimate the probabilities.

The use of EM algorithm in this work involves an iterative process with two alternating steps:

1. An expectation step (E-step), where posterior probabilities for the latent variables (in our case, the variable z_k) are computed, based on the current estimates of the parameters;
2. A maximization step (M-step), where estimates for the parameters are updated to maximize the complete data likelihood.

After proper initialization of the parameters including $P(z_k)$, $P(w|z_k)$, and $P(d|z_k)$, the algorithm alternates between the following two steps before a local optimal solution is reached.

– in E-step, we compute

$$P(z_k|d, w) = \frac{P(z_k)\,P(d|z_k)\,P(w|z_k)}{\sum_{z'_k \in \mathcal{Z}} P\left(z'_k\right) P\left(d|z'_k\right) P\left(w|z'_k\right)};$$

– in M-step, we update the model parameters with

$$P(w|z_k) = \frac{\sum_{d \in \mathcal{D}} c(d, w)\,P(z_k|d, w)}{\sum_{d \in \mathcal{D}} \sum_{w' \in \mathcal{W}} c(d, w')\,P(z_k|d, w')},$$

$$P(d|z_k) = \frac{\sum_{w \in \mathcal{W}} c(d, w)\,P(z_k|d, w)}{\sum_{d' \in \mathcal{D}} \sum_{w \in \mathcal{W}} c(d', w)\,P(z_k|d', w)},$$

and

$$P(z_k) = \frac{\sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} c(d, w)\,P(z_k|d, w)}{\sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} c(d, w)}.$$

Once the model is constructed, we can calculate the posterior probability $P(z_k|d)$ using the Bayes rule:

$$P(z_k|d) = \frac{P(d|z_k)\,P(z_k)}{\sum_{z_k \in \mathcal{Z}} P(d|z_k)\,P(z_k)}.$$

Here $P(z_k|d)$ can be interpreted as how much a hidden sentiment factor $z_k(\in \mathcal{Z})$ "contributes" to the review document d, and probabilities $\{P(z_k|d)|z_k \in \mathcal{Z}\}$ can be considered as a summarization of d in terms of sentiments.

4 Adaptive sentiment analysis models

To make a sentiment analysis model adaptive to changes as new reviews become available, a naïve way is to re-train the model from scratch using all data available including the newly obtained data. This has two drawbacks: (i) it is clearly highly inefficient, especially when the data volume is high; and (ii) the out-of-date reviews from a long time ago and not relevant anymore may actually harm the performance of the model if they are included in training. An alternative solution is to develop a method that only takes the newly available data into consideration and discards all old data. This approach, however, may suffer from the problem of not having sufficient amount of training samples, as it is very likely that only a few reviews are written within a short period of time. Also, discarding the old data in its entirety may be unwise, because knowledge obtained from those data (which is reflected in the model parameters) is lost.

We propose two methods based on S-PLSA, denoted S-PLSA* and S-PLSA$^+$ respectively. Both of them conduct adaptive sentiment analysis by automatically adjusting model parameters over time without incurring too much computational cost.

4.1 Light-weight incremental model

When the characteristics of the underlying data are relatively stable and do not evolve significantly over time, it is possible to train the S-PLSA model in a batch manner on a collection of reviews, and apply the trained model on unseen reviews encountered in the future. As described in Section 3, the process of learning parameters in a S-PLSA model is divided into two steps: the E-step estimates the objective function $P(z_k|d, w)$ for each document-word pair in the training data, and the M-step updates unknown parameters $P(z_k)$, $P(w|z_k)$, and $P(d|z_k)$ for each latent sentiment factor. From the M-step equations in Section 3, we notice that if the number of documents N is sufficiently large, the addition of a new document is unlikely to have significant influence on parameter estimation. Therefore, when a new review document v becomes available, it is possible to update the parameters incrementally (though approximately) without invoking a reconstruction of the model, and the parameter values thus obtained are expected to be close to those obtained by re-training the model from scratch. Based on this observation, we develop an adaptive method called S-PLSA*, which can be considered as a simplified method for parameter estimation in the EM algorithm.

S-PLSA* works as follows. When a new document v becomes available, we first calculate $P(z_k|v, w)$ for latent sentiment factor z_k under the existing models. Then, the model parameters $P(z_k)$, $P(w|z_k)$, and $P(d|z_k)$ are incrementally updated as illustrated in Algorithm 1. The computation in S-PLSA* involves only the new data

and the current parameters in the model. It does not require any iteration, and is independent of the old data. Therefore, it consumes very little time and space, and is orders of magnitude faster than overhaul re-training on the whole collection of data.

Algorithm 1: Adaptive S-PLSA* with a new review

Input: θ and newly available data v
Output: updated θ

1 **for** $k = 1$ *to* K **do**
2 $\quad P(z_k|v,w) = \frac{P(z_k)P(v|z_k)P(w|z_k)}{\sum_{z'_k \in \mathcal{Z}} P(z'_k)P(v|z'_k)P(w|z'_k)}$;
3 **end**
4 **for** $k = 1$ *to* K **do**
5 $\quad P(w|z_k) = \frac{\sum_{d\in\mathcal{D}} c(d,w)P(z_k|d,w)+c(v,w)P(z_k|v,w)}{\sum_{d\in\mathcal{D}}\sum_{w'\in\mathcal{W}} c(d,w')P(z_k|d,w')+\sum_{w'\in\mathcal{W}} c(v,w')P(z_k|v,w')}$,
6 **end**
7 **for** $k = 1$ *to* K **do**
8 $\quad P(d|z_k) = \frac{\sum_{w\in\mathcal{W}} c(d,w)P(z_k|d,w)}{\sum_{d'\in\mathcal{D}}\sum_{w\in\mathcal{W}} c(d',w)P(z_k|d',w)+\sum_{w\in\mathcal{W}} c(v,w)P(z_k|v,w)}$,
9 **end**
10 **for** $k = 1$ *to* K **do**
11 $\quad P(z_k) = \frac{\sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} c(d,w)P(z_k|d,w)+\sum_{w\in\mathcal{W}} c(v,w)P(z_k|v,w)}{\sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} c(d,w)+\sum_{w\in\mathcal{W}} c(v,w)}$.
12 **end**
13 return updated θ

In Algorithm 1, the update of the objective function is described in lines 1–3, and parameter re-estimation is presented from line 4 to line 12.

Compared with the original method for parameter estimation, where $P(w|z_k) = \frac{\sum_{d\in\mathcal{D}} c(d,w)P(z_k|d,w)}{\sum_{d\in\mathcal{D}}\sum_{w'\in\mathcal{W}} c(d,w')P(z_k|d,w')}$, $P(d|z_k) = \frac{\sum_{w\in\mathcal{W}} c(d,w)P(z_k|d,w)}{\sum_{d'\in\mathcal{D}}\sum_{w\in\mathcal{W}} c(d',w)P(z_k|d',w)}$, and $P(z_k) = \frac{\sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} c(d,w)P(z_k|d,w)}{\sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} c(d,w)}$, the set of updating equations in Algorithm 1 reflect the changes of weights in each component when a new review becomes available.

As described in [18], the time complexity of the standard PLSA is $O(K \times N)$ per iteration in EM, where K is the number of categories and N is the number of term-document co-occurrences. In the worst case, N is equal to $M \times (m + n)$ where M is the total number of sentiment words, while m and n are the total numbers of reviews in the training dataset and the testing dataset, respectively. Assuming that it takes I iterations for the EM algorithm to converge, the total cost is $O(I(K \times N))$. In real applications, where M, m, and n are usually large, it may take very long for the algorithm to converge.

In contrast, S-PLSA* overcomes this obstacle by executing only one iteration, and the overall complexity therefore comes down to $O(K \times N)$. Moreover, observing Algorithm 1, we notice that if we maintains a record of the values of every $P(z_k|v, w)$,

$P(w|z_k)$, $P(d|z_k)$, and $P(z_k)$ in the model, only the new probabilities have to be computed in the current re-estimation. In this way, the runtime cost can be further reduced to $O(K \times L)$, where L is the number of new reviews.

4.2 Quasi-Bayesian model

Although highly efficient, the S-PLSA* method proposed in Section 4.1 may degrade in accuracy over time, as the errors incurred by the approximation may accumulate as more and more reviews are added. Moreover, this method does not retire or reduce the influence of the out-dated (and thus maybe irrelevant) reviews, which may further affect the quality of the model. We thus hope to develop a more sophisticated method that could mitigate those problems. The basic idea is to perform update using the newly available reviews and fade away the out-dated data at the same time by:

1. incrementally accumulating statistics on the training data, and
2. fading out the out-of-date data.

In this section, we propose S-PLSA$^+$, which applies the quasi-Bayesian incremental learning method proposed in [6].

Let $\mathcal{D}_n$ be the set of reviews made available at epoch n (e.g., the reviews published on a certain day, but the time unit used can be set to be finer or coarser based on the need), and denote by $\chi^n = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$ the set of reviews obtained up to epoch n. In order to support parameter update based on the new data, we take a Bayesian approach and perform maximum a posteriori (MAP) estimation instead of maximum-likelihood estimation as stated in Section 3. The MAP estimates for S-PLSA$^+$at epoch n are determined by maximizing the posterior probability using χ^n:

$$\theta^{(n)} = \arg\max_{\theta} P\left(\theta|\chi^n\right) = \arg\max_{\theta} P(\mathcal{D}_n|\theta)\, P\left(\theta|\chi^{n-1}\right) \tag{4}$$

The learning (i.e., update of parameters) is expected to be done repeatedly at different epochs.

In order to allow closed-form recursive update of θ, we use the closest tractable parametric prior distribution $g(\theta|\phi^{(n-1)})$ with sufficient statistics to approximate the posterior distribution $P(\theta|\chi^{n-1})$, where ϕ^{n-1} is evolved from review sets χ^{n-1}. This leads to

$$\theta^{(n)} \approx \arg\max_{\theta} P(\mathcal{D}_n|\theta) g\left(\theta|\phi^{(n-1)}\right). \tag{5}$$

Note that at epoch n, only the new reviews $\mathcal{D}_n$ and the current statistics $\phi^{(n-1)}$ are used to update the S-PLSA$^+$ parameters, and the set of reviews $\mathcal{D}_n$ are discarded after new parameter values $\phi^{(n)}$ are obtained, which results in significant savings in computational resources.

The particular choice of the prior $g(\theta|\phi)$ in our model is the Dirichlet distribution, which can be expressed by

$$g(\theta|\phi) = \prod_{k=1}^{K} \left[\prod_{j=1}^{M} P(w_j|z_k)^{\alpha_{j,k}-1} \prod_{i=1}^{N} P(z_k|d_i)^{\beta_{k,i}-1} \right] \tag{6}$$

where $\phi = \{\alpha_{j,k}, \beta_{k,i}\}$ are the hyperparameters of the Dirichlet distribution. This choice of conjugate prior allows for a closed-form solution for fast model adaptation.

Assuming for the moment that $\phi^{(n-1)}$ is known, we can show that $\theta^{(n)}$ can be obtained through an EM algorithm [6], and $\theta^{(n)}$ can be obtained by

$$\hat{P}^{(n)}\left(w_j^{(n)}|z_k\right) = \frac{\sum_{i=1}^{N} c\left(d_i^{(n)}, w_j^{(n)}\right) P\left(z_k|d_i^{(n)}, w_j^{(n)}\right) + \left(\alpha_{j,k}^{(n-1)} - 1\right)}{\sum_{m=1}^{M}\left[\sum_{i=1}^{N} c\left(d_i^{(n)}, w_m^{(n)}\right) P\left(z_k|d_i^{(n)}, w_m^{(n)}\right) + \left(\alpha_{j,m}^{(n-1)} - 1\right)\right]} \tag{7}$$

$$\hat{P}^{(n)}\left(z_k|d_i^{(n)}\right) = \frac{\sum_{j=1}^{M} c\left(d_i^{(n)}, w_j^{(n)}\right) P\left(z_|d_i^{(n)}, w_j^{(n)}\right) + \left(\beta_{k,i}^{(n-1)} - 1\right)}{c\left(d_i^{(n)}\right) + \sum_{l=1}^{K}\left(\beta_{l,i}^{(n-1)} - 1\right)}. \tag{8}$$

A major benefit of S-PLSA$^+$ lies in its ability to continuously update the hyperparameters. We can show that the new hyperparameters are given by

$$\alpha_{j,k}^{(n)} = \sum_{i=1}^{|\mathcal{D}_n|} c\left(d_i^{(n)}, w_j^{(n)}\right) P^{(n)}\left(z_k|d_i^{(n)}, w_j^{(n)}\right) + \alpha_{j,k}^{(n-1)} \tag{9}$$

$$\beta_{k,i}^{(n)} = \sum_{j=1}^{M} c\left(d_i^{(n)}, w_j^{(n)}\right) P^{(n)}\left(z_k|d_i^{(n)}, w_j^{(n)}\right) + \beta_{k,i}^{(n-1)}. \tag{10}$$

where the posterior $P^{(n)}(z_k|d_i^{(n)}, w_j^{(n)})$ is computed using $\mathcal{D}_n$ and the current parameters $\theta^{(n)}$, and $c(d_i^{(n)}, w_j^{(n)})$ denotes the number of $(d_i^{(n)}, w_j^{(n)})$ pairs.

To summarize, S-PLSA$^+$ works as follows. In the startup phase, initial estimates of the hyperparameters $\phi^{(0)}$ are obtained. Then, at each learning epoch n, (i) new estimates of the parameters $\theta^{(n)}$ are computed based on the newly available data $\mathcal{D}_n$ and hyperparameters obtained from epoch $n-1$; and (ii) new estimates of the hyperparameters $\phi^{(n)}$ are obtained using (9) and (10). In this way, the model is continuously updated when new reviews ($\mathcal{D}_n$) become available, and at the same time fades out historical data χ^{n-1}, with the information contained in χ^{n-1} already

captured by $\phi^{(n-1)}$. The implementation procedure for parameter update at each epoch n is shown in Algorithm 2.

Algorithm 2: S-PLSA$^+$ parameter update at epoch n

Input: $\theta^{(n-1)}$, newly available data $\mathcal{D}_n$, hyperparameters $\phi^{(n-1)}$, and EM algorithm threshold ϵ

Output: $\theta^{(n)}$, $\phi^{(n)}$

while $\left|\frac{\theta^{(n)}-\theta^{(n-1)}}{\theta^{(n)}}\right| > \epsilon$ **do**

  compute $\hat{P}^{(n)}(w_j^{(n)}|z_k) = \frac{\sum_{i=1}^{N} c(d_i^{(n)}, w_j^{(n)}) P(z_k|d_i^{(n)}, w_j^{(n)}) + (\alpha_{j,k}^{(n-1)} - 1)}{\sum_{m=1}^{M}\left[\sum_{i=1}^{N} c(d(n)_i, w_m^{(n)}) P(z_k|d_i^{(n)}, w_m^{(n)}) + (\alpha_{j,m}^{(n-1)} - 1)\right]}$;

  compute $\hat{P}^{(n)}(z_k|d_i^{(n)}) = \frac{\sum_{j=1}^{M} c(d_i^{(n)}, w_j^{(n)}) P(z_|d_i^{(n)}, w_j^{(n)}) + (\beta_{k,i}^{(n-1)} - 1)}{c(d_i^{(n)}) + \sum_{l=1}^{K} (\beta_{l,i}^{(n-1)} - 1)}$;

end

update $\alpha_{j,k}^{(n)} = \sum_{i=1}^{|\mathcal{D}_n|} c(d_i^{(n)}, w_j^{(n)}) P^{(n)}(z_k|d_i^{(n)}, w_j^{(n)}) + \alpha_{j,k}^{(n-1)}$;

update $\beta_{k,i}^{(n)} = \sum_{j=1}^{M} c(d_i^{(n)}, w_j^{(n)}) P^{(n)}(z_k|d_i^{(n)}, w_j^{(n)}) + \beta_{k,i}^{(n-1)}$;

return $\theta^{(n)}$, $\phi^{(n)}$;

5 Application to sales prediction

The proposed adaptive models can be employed in a variety of tasks to reflect the sentiment changes as time evolves, e.g., sentiment clustering, sentiment classification, etc. As a sample application, we plug it into the ARSA model proposed in [23], and then use it to predict sales performance based on reviews and past sales figures.

5.1 Autoregressive sentiment aware model

The ARSA model [23] aims to capture two different factors that can help predict the current product sales. One factor is the revenues from the preceding period. Naturally, the sales performance of the current period is strongly related to that of the immediately preceding period. The other factor we consider is people's sentiments about the product, reflected in online reviews.

In ARSA, the temporal relationship between the product sales of the preceding periods (say, days) and the current period (day) is modeled by an autoregressive (AR) process. An AR model is a linear model that aims at predicting an output y_n of a system based on previous outputs $(y_{n-1}, y_{n-2}, \ldots)$ and inputs $(x_n, x_{n-1}, x_{n-2}, \ldots)$. Let the sales of the product at day t be x_t $(t = 1, 2, \ldots, N$ where $t = 1$ and $t = N$ correspond to the first and last day of interest), and $\{x_t\}(t = 1, \ldots, N)$ denote the time series $x_1, x_2, \ldots, x_N$. Then an AR process of order p can be formulated as

$$x_t = \sum_{i=1}^{p} \phi_i x_{t-i} + \epsilon_t, \tag{11}$$

where $\phi_1, \phi_2, \ldots, \phi_p$ are model parameters, and ϵ_t is an error term (white noise with zero mean).

In ARSA, this model is further extended to take sentiments into consideration, because the product sales might be greatly influenced by people's sentiments in the same time period.

Let $\mathcal{D}_t$ be the set of reviews on the product of interest that were posted on day t, and $p(z = j|d)(d \in \mathcal{D}_t)$ be the probability of sentiment factor $z = j$ conditional on review d according to the S-PLSA model. The ARSA model is formulated as follows.

$$y_t = \sum_{i=1}^{p} \varphi_i y_{t-i} + \sum_{i=1}^{q} \sum_{j=1}^{R} \rho_{i,j} \omega_{t-i,j} + \epsilon_t, \tag{12}$$

where

1. y_t denotes the sales figure at time t after proper pre-processing such as de-seasoning,
2. p, q, and R are user-chosen parameters,
3. φ_i and $\rho_{i,j}$ are coefficients to be estimated using training data, and
4. $\omega_{t,j} = \frac{1}{|\mathcal{D}_t|} \sum_{d \in \mathcal{D}_t} p(z_j|d)$, where $\mathcal{D}_t$ is the set of reviews available at time t and $p(z_j|d)$ is computed based on S-PLSA$^+$ or S-PLSA*. Intuitively, $\omega_{t,j}$ represents the average fraction of the sentiment that can be attributed to the hidden sentiment factor j within $\mathcal{D}_t$.

As such, the ARSA model not only considers the influence of past product sales, but also takes into account the sentiment information distilled from the reviews.

The set of parameters to be learned in ARSA include $\phi_i (i = 1, \ldots, p)$, and $\rho_{i,j} (i = 1, \ldots, q;\ j = 1, \ldots, K)$, and $\omega_{t,j}$. Parameter estimation can be done through linear least squares fitting when p and q are fixed.

Let $\boldsymbol{\alpha}_{m,t} = (y_{m,t-1}, \ldots, y_{m,t-p}, \omega_{m,t-1,1}, \ldots, \omega_{m,t-q,k})^T$, where the subscript m ($1 \leq m \leq M$, M is the number of distinct products) is used to refer to a particular product, and $y_{m,t}$ corresponds the sales quantity for product m at time t. Then (12) can be rewritten as

$$\boldsymbol{\alpha}_{m,t}^T \boldsymbol{\theta} = y_{m,t}.$$

Let $A = (\boldsymbol{\alpha}_{1,1}, \boldsymbol{\alpha}_{1,2}, \ldots)^T$, and $\mathbf{c} = (y_{1,1}, y_{1,2}, \ldots)$. Then based on the training data, we seek to find a solution $\hat{\boldsymbol{\theta}}$ for the "equation"

$$A\boldsymbol{\theta} \approx \mathbf{c}.$$

Apparently, this is a least squares regression problem that be solved using standard techniques in mathematics.

Note that the notion of time (t) in the ARSA model is different from the epoch (n) in S-PLSA$^+$ and S-PLSA*. For example, sales prediction can be made for each day using ARSA, whereas the model adaptation of S-PLSA$^+$ can happen every other day.

5.2 Implementation issues

It is important to note that AR models are only appropriate for time series that are stationary, which can hardly be true in practice. For example, retail sales tend to peak for the Christmas season and then decline after the holidays. Thus, time series of retail sales will typically show an increasing trend from September to December and a declining trend from January to February. In addition, a newly released product tends to garner more attention before or close to its release date due to various reasons, such as aggressive marketing, unique features, or being controversial. This

may temporally boost the product's sales performance for a short period of time. But as time goes by, the discussion over this product is likely to fade out. Therefore, to accurately predict the future product sales, it is necessary to take those trends or "seasonalities" into consideration.

As a case study, we investigate the important problem of predicting future box office revenues in the movie sector. The choice of using movies rather than other products in our study is mainly due to data availability. The daily box office revenue data are all published on the Web and readily available, unlike other product sales data that are often private to their respective companies due to obvious reasons. Also, as discussed by Liu et al. [21], analyzing movie reviews is one of the most challenging tasks in sentiment mining. We expect the models and algorithms developed for box office prediction to be easily adapted to handle other types of products that are subject to online discussions, such as books, music CDs and electronics. Apparently, for box office prediction, the time series $\{x_t\}$ in (11) are not stationary, because there normally exist clear trends and "seasonalities" in the series. Based on observations in [23], there is a negative exponential downward trend for the box office revenues as the time moves further from the release date. Seasonality is also present, as within each week, the box office revenues always peak at the weekend and are generally lower during weekdays. Therefore, in order to properly model the time series $\{x_t\}$, some preprocessing steps are required.

The first step is to remove the trend. This is achieved by first transforming the time series $\{x_t\}$ into the logarithmic domain, and then differencing the resulting time series $\{x_t\}$. The new time series obtained is thus

$$x'_t = \Delta \log x_t = \log x_t - \log x_{t-1}.$$

We then proceed to remove the seasonality [10]. To this end, we apply the lag operator on $\{x'_t\}$ and obtain a new time series $\{y_t\}$ as follows:

$$y_t = x'_t - L^7 x' t = x'_t - x'_{t-7}.$$

By computing the difference between the box office revenue of a particular date and that of seven days ago (the lag is seven days), we effectively removed the seasonality factor due to different days of a week. After the preprocessing step, a new AR model can be formed on the resulting time series $\{y_t\}$:

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t. \tag{13}$$

After that, y_t derived from (13) can be plugged into (12) to estimate the temporal relationship between the future product sales and those of the preceding days.

Besides, it is worth noting that although the AR model developed here is specific for movies, the same methodology can be applied in other contexts. For example, trends and seasonalities are present in the sales performance of many different products (such as electronics and music CDs, where the corresponding spans of the lag can be a month or a year). Therefore the preprocessing steps described above to remove them can be adapted and used in the predicting the sales performance.

6 Experiments

Experiments are conducted on an IMDB dataset to evaluate the effectiveness of the proposed adaptive models, and the prediction power of ARSA using these methods. We first describe the dataset used in the experiments, and then present the experimental results.

6.1 Data description

The dataset was obtained from the IMDB Website[1] and have two parts. Part 1, denoted by IMDB-REVIEW, consists of 28,353 reviews for 20 drama films released in the US from 1 May 2006 to 1 September 2006, and Part 2, denoted by IMDB-BO, contains the daily gross box office revenues of those films. For each review, we extracted the title, free text contents, time stamp, etc., and then indexed them using Apache Lucene.[2] We used the MATLAB environment to program the proposed methods, and the experiments were performed on a SUN Sun Fire V440 Server with a 1.3 GHz UltraSparc 3i Processor, and 16GB RAM.

6.2 Perplexity evaluation

We first evaluate the effectiveness of the proposed adaptive models by computing their perplexity on the IMDB-REVIEW dataset. Perplexity is a commonly used measure of goodness for statistical language models. It is defined as the inverse of the probability of the test set as assigned by the language model, normalized by the number of words. Roughly speaking, it corresponds to the weighted average word branching factor of a language model. Lower perplexity indicates better modeling capability of the model on the given corpus (dataset).

In the PLSA-based models, there are several user-chosen parameters that provide the flexibility to fine tune the model for optimal performance. One of those parameters is the number of hidden sentiment factors, K. In order to study how K affects the prediction accuracy, we conducted empirical studies by varying its values from 2 to 15. Our results indicate that the best performance is achieved at $K = 4$ with our movie data. Hence, we set the number of latent factors in the original PLSA model and the adaptive models to 4, and compared their perplexities. As discussed in preceding sections, only appraisal words are employed to construct the feature vectors used in those models.

We first use the reviews from IMDB-REVIEW that correspond to 10 randomly chosen films to train an S-PLSA model. This model is then adapted using the S-PLSA$^+$ in four epochs, with one-fourth of the remaining reviews used as adaptation reviews at each epoch. For S-PLSA*, we use the same reviews at each epoch as those for S-PLSA$^+$ to adapt the model accordingly. In particular, we set epochs on $d = (5, 10, 15, 20)$ with reviews available upto day d for both S-PLSA* and S-PLSA$^+$. In addition, for the original S-PLSA model, we perform parameter estimation for all the reviews made available at each epoch $i = (1, 2, 3, 4)$. We perform 10-fold validation over training and adaptation sets.

[1] http://www.imdb.com

[2] http://lucene.apache.org

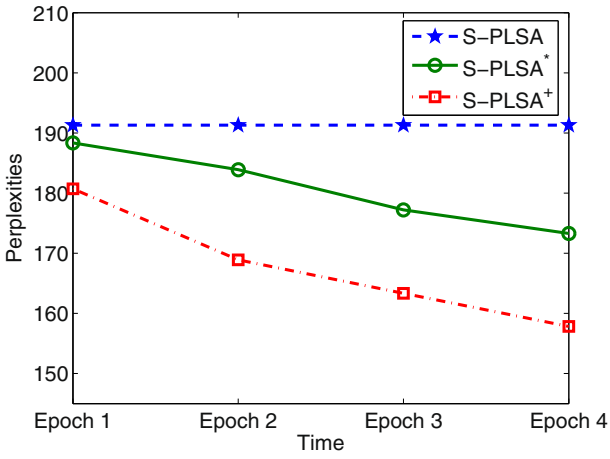

Figure 1 Perplexities for S-PLSA, S-PLSA*, and S-PLSA$^+$ at different epochs.

Figure 1 shows the perplexities of the original S-PLSA model and the adaptive models at different adaptation epochs with the number of latent factors $K = 4$. It is apparent from the graph that the perplexities are consistently reduced with incremental model adaptation as more adaptation data are introduced at each epoch. This testifies to the fact that model adaptation does help the sentiment modeling of the reviews.

6.3 Efficiency evaluation

In this section, we measure the construction time of the S-PLSA* and S-PLSA$^+$ models as we vary the size of the data sets accumulated at different epoches, and compare the results with that of reconstructing a S-PLSA every time from scratch. Similar to the experiment described in Section 6.2, we use the same reviews for both S-PLSA* and S-PLSA$^+$ at each epoch. In addition, we set epoches on $d = (5, 10, 15, 20)$ with reviews available upto day d. For the original S-PLSA model, we perform parameter estimation for all the reviews made available at each epoch $i = (1, 2, 3, 4)$, and the EM algorithm is run until a convergence threshold of 0.0001 is reached. We record the elapsed time for each method at each epoch, and the results are reported in Table 1.

Clearly, the proposed S-PLSA$^+$ and S-PLSA* have an additional benefit of being much faster because the methods do not require all paremeters to converge, as is usually needed by the EM algorithm. In addition, the light-weight adaptive method S-PLSA* requires even less time, which justifies our analysis in Section 4.1.

Table 1 Comparison of training/adapting time (seconds).

Epoch	1	2	3	4
S-PLSA$^+$	2.16	2.14	2.29	2.31
S-PLSA*	0.114	0.128	0.117	0.156
S-PLSA	617.21	1287.68	2112.05	2598.34

6.4 Effectiveness for sales prediction

The effectiveness of our adaptive methods for sales performance prediction is evaluated by replacing the S-PLSA component in the original ARSA model. Like in the previous perplexity experiments, reviews for half of the movies are used for batch training. For the original ARSA, the trained model is then used to make predictions in the testing data consisting of the other half the movies. For the proposed model, adaptation of the S-PLSA* and S-PLSA$^+$ components is performed for each movie in the testing set, in four epochs on four different days d ($d = 2, 4, 6, 8$) using the review data available up to day d. The up-to-date model at day d is then used for subsequent prediction tasks.

The mean absolute percentage error (MAPE) is used to measure the prediction accuracy:

$$MAPE = \frac{1}{T}\sum_{i=1}^{T}(|Pred_i - True_i|/True_i), \tag{14}$$

where T is the number of instances in the testing set, and $Pred_i$ and $True_i$ are the predicted value and the true value respectively.

Figure 2 shows the MAPE of the original ARSA with S-PLSA, and the ARSA using S-PLSA* and S-PLSA$^+$ updated at Epochs 1–4 ($d = 2, 4, 6, 8$). It is apparent from the figure that accuracy of S-PLSA$^+$ is superior to that of the other two approaches. The accuracy of the model improves significantly as the it is getting updated in the first two epochs, which demonstrates the benefits of having an incremental model to absorb new information; especially in our case, S-PLSA$^+$ allows the models to be adapted to the individual movies. The rate of increase in accuracy get slower from Epoch 2 through Epoch 4, indicating that no significant new information is available from Epoch 2 to Epoch 4. The proposed model also outperforms the S-PLSA* approach where a rough and quick re-estimation of system parameters is completed at each epoch.

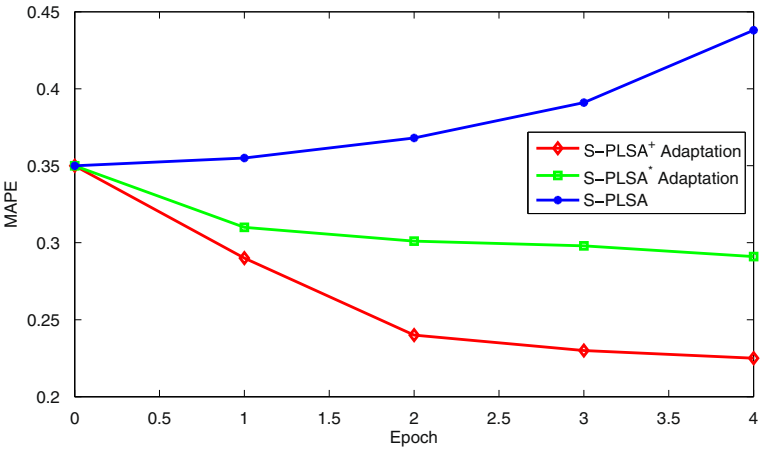

Figure 2 The MAPE of ARSA with S-PLSA, S-PLSA*, and S-PLSA$^+$ at different epochs.

6.5 Comparison with alternative methods

In the absence of prior work that deals with the problem of adaptive sentiment analysis, we devise a few possible alternatives to our proposed methods as the baseline for comparison.

Again, we adopt the ARSA model to compare the effectiveness of our proposed models. Similar to the previous experiment in Section 6.4, the prediction performance of the alternative method for sales prediction is obtained by replacing the S-PLSA component in the ARSA model. However, in this experiment, we set epochs on four different days d ($d = 5, 10, 15, 20$) using reviews available up to day d. Our purpose of setting a longer period between two epochs is to examine the effectiveness of two adaptation algorithms in the long run.

We have designed two strategies for comparison. First, to verify that the adaptive sentiment information captured by the S-PLSA$^+$ model plays an important role in box office revenue prediction, we compare ARSA with adaptation to non-PLSA-based method which do not take sentiment information into consideration. To this end, we conduct experiments to compare ARSA against the pure autoregressive (AR) model without any terms on sentiments, i.e., $y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t$.

We then compare the proposed methods with another alternative where the S-PLSA component is completely re-trained from scratch using all data available including the original training data. This represents a batch adaptation approach. The results are shown in Figure 3.

It is clear that the proposed S-PLSA$^+$ still enjoys the best performance compared to the other three methods. Compared to the original S-PLSA model, the performance of S-PLSA* degrades slowly. Nonetheless, its performance is still not comparable to that of S-PLSA$^+$ due to the approximate nature of this solution. If there is no significant change in the review contents between epochs, this alternative method may have a slower rate of performance degradation. S-PLSA$^+$ even outperform the re-training approach where the S-PLSA is completely retrained at each epoch. This is due to the fact that some information in the original training set may be out-of-date

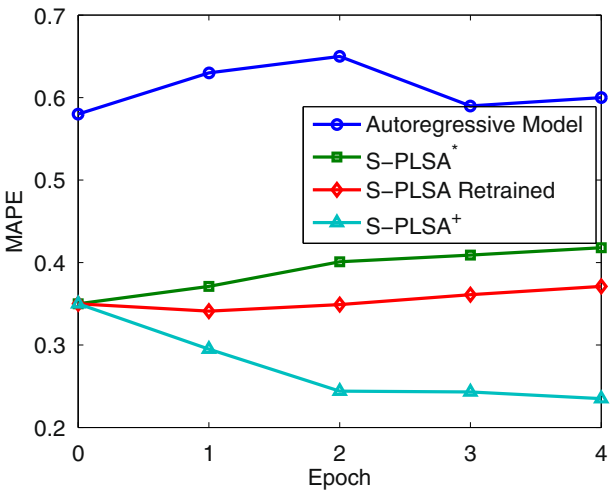

Figure 3 The MAPE of ARSA with S-PLSA, ARSA with S-PLSA$^+$+, and ARSA with alternative adaptation method at different epochs.

and not as relevant as the newly available reviews that focus more on the individual movies that we are making the prediction for. The proposed model can discount such out-of-date and irrelevant information, whereas the re-training approach cannot.

7 Conclusions and future work

In this paper, we have presented two adaptive methods that are capable of incrementally updating the parameters of the S-PLSA model when new review data become available. They have been used in conjunction with the ARSA model for predicting sales performance. Experimental results on a movie dataset show that by allowing the model to be adaptive, we can capture new sentiment changes arising from newly available reviews, which can greatly improve its modeling capability as well as the accuracy when used for prediction. For future work, we plan to study the performance of these models in other information retrieval and data mining tasks.

Acknowledgements This work was supported in part by National Natural Science Foundation of China Grants (No. 61070018, No. 60903108), the Program for New Century Excellent Talents in University (NCET-10-0532), NSERC Discovery Grants, an Early Career Award of Ontario, the Independent Innovation Foundation of Shandong University (2009TB016,2012ZD12), and the SAICT Experts Program.

References

1. Archak, N., Ghose, A., Ipeirotis, P.G.: Show me the money!: deriving the pricing power of product features by mining consumer reviews. In: KDD, pp. 56–65 (2007)
2. Basilico, J., Hofmann, T.: Unifying collaborative and content-based filtering. In: ICML, pp. 65–72 (2004)
3. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Boykin, S., Merlino, A.: Machine learning of event segmentation for news on demand. Commun. ACM **43**(2), 35–41 (2000)
5. Chevalier, J.A., Mayzlin, D.: The effect of word of mouth on sales: online book reviews. J. Mark. Res. **43**(3), 345–354 (2006)
6. Chien, J.-T., Wu, M.-S.: Adaptive bayesian latent semantic analysis. IEEE Trans. Audio Speech Lang. Processing **16**(1), 198–207 (2008)
7. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: WWW, pp. 519–528 (2003)
8. Dellarocas, C., Awad, N.F., Zhang, X.(Michael): Exploring the value of online reviews to organizations: implications for revenue forecasting and planning. In: ICIS, pp. 379–386 (2004)
9. Dellarocas, C., Zhang, X.(Michael), Awad, N.F.: Exploring the value of online product ratings in revenue forecasting: the case of motion pictures. J. Interact. Market **21**(4), 23–45 (2007)
10. Enders, W.: Applied Econometric Time Series, 2nd edn. Wiley, New York (2004)
11. Forman, C., Ghose, A., Wiesenfeld, B.: Examining the relationship between reviews and sales: the role of reviewer identity disclosure in electronic markets. Inf. Syst. Res **19**(3), 291–313 (2008); Special issue on the interplay between digital and social networks
12. Ghose, A., Ipeirotis, P.G.: Designing novel review ranking systems: predicting the usefulness and impact of reviews. In: ICEC, pp. 303–310 (2007)
13. Ghose, A., Ipeirotis, P.G.: Estimating the helpfulness and economic impact of product reviews: mining text and reviewer characteristics. IEEE Trans. Knowl. Data Eng. **23**(10), 1498–1512 (2011)
14. Girolami, M., Kabán, A.: On an equivalence between plsi and lda. In: SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, pp. 433–434 (2003)

15. Grefenstette, G., Qu, Y., Shanahan, J.G., Evans, D.A.: Coupling niche browsers and affect analysis for an opinion mining application. In: Proceedings of RIAO-04, 7th International Conference on Recherche d'Information Assiste par Ordinateur, pp. 186–194 (2004)
16. Gruhl, D., Guha, R., Kumar, R., Novak, J., Tomkins, A.: The predictive power of online chatter. In: KDD '05, pp. 78–87 (2005)
17. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: WWW '04, pp. 491–501 (2004)
18. Hofmann, T.: Probabilistic latent semantic analysis. In: UAI'99, pp. 289–296 (1999)
19. Hofmann, T., Puzicha, J.: Latent class models for collaborative filtering. In: IJCAI, pp. 688–693 (1999)
20. Kamps, J., Marx, M.: Words with attitude. In: Proc. of the First International Conference on Global WordNet, pp. 332–341 (2002)
21. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: WWW, pp. 342–351 (2005)
22. Liu, Y., Huang, X., An, A.: Personalized recommendation with adaptive mixture of Markov models. J. Am. Soc. Inf. Sci. Technol. **58**(12), 1851–1870 (2007)
23. Liu, Y., Huang, X., An, A., Yu, X.: ARSA: a sentiment-aware model for predicting sales performance using blogs. In: SIGIR, pp. 607–614 (2007)
24. Morinaga, S., Yamanishi, K., Tateishi, K., Fukushima, T.: Mining product reputations on the web. In: KDD '02: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 341–349 (2002)
25. Nasukawa, T., Yi, J.: Sentiment analysis: capturing favorability using natural language processing. In: K-CAP '03: Proceedings of the 2nd International Conference on Knowledge Capture, pp. 70–77 (2003)
26. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: ACL '04, pp. 271–278 (2004)
27. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: ACL '05, pp. 115–124 (2005)
28. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: EMNLP, pp. 79–86 (2002)
29. Pavlou, P.A., Dimoka, A.: The nature and role of feedback text comments in online marketplaces: implications for trust building, price premiums, and seller differentiation. Inf. Syst. Res **17**(4), 392–414 (2006)
30. Rosen, S.: Hedonic prices and implicit markets: product differentiation in pure competition. J. Polit. Econ. **82**(1), 34–55 (1974)
31. Rubicon Consulting Inc. Online communities and their impact on business: ignore at your peril, October (2008)
32. Snyder, B., Barzilay, R.: Multiple aspect ranking using the good grief algorithm. HLT-NAACL, pp. 300–307 (2007)
33. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: ACL '02, pp. 417–424 (2001)
34. Wang, H., Zhang, D., Zhai, C.: Structural topic model for latent topical structure analysis. In: ACL, pp. 1526–1535 (2011)
35. Whitelaw, C., Garg, N., Argamon, S.: Using appraisal groups for sentiment analysis. In: CIKM '05, pp. 625–631 (2005)
36. Wiebe, J., Wilson, T., Bruce, R., Bell, M., Martin, M.: Learning subjective language. Comput. Linguist. **30**(3), 277–308 (2004)
37. Yu, H., Hatzivassiloglou, V.: Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 129–136 (2003)
38. Zhang, Z., Varadarajan, B.: Utility scoring of product reviews. In: CIKM, pp. 51–57 (2006)
39. Zhong, N., Liu, J., Yao, Y.Y.: In search of the wisdom web. Computer **35**(11), 27–31 (2002)
40. Zhong, N., Liu, J., Yao, Y.: Envisioning intelligent information technologies through the prism of web intelligence. Commun. ACM **50**(3), 89–94 (2007)
41. Zhong, N., Ma, J., Huang, R., Liu, J., Yao, Y., Zhang, Y., Chen, J.: Research challenges and perspectives on wisdom web of things (w2t). J. Supercomput 1–21 (2010). doi:10.1007/s11227-010-0518-8
42. Zhu, F., Zhang, X.(Michael): The influence of online consumer reviews on the demand for experience goods: the case of video games. In: International Conference on Information Systems (ICIS), pp. 27–51 (2006)